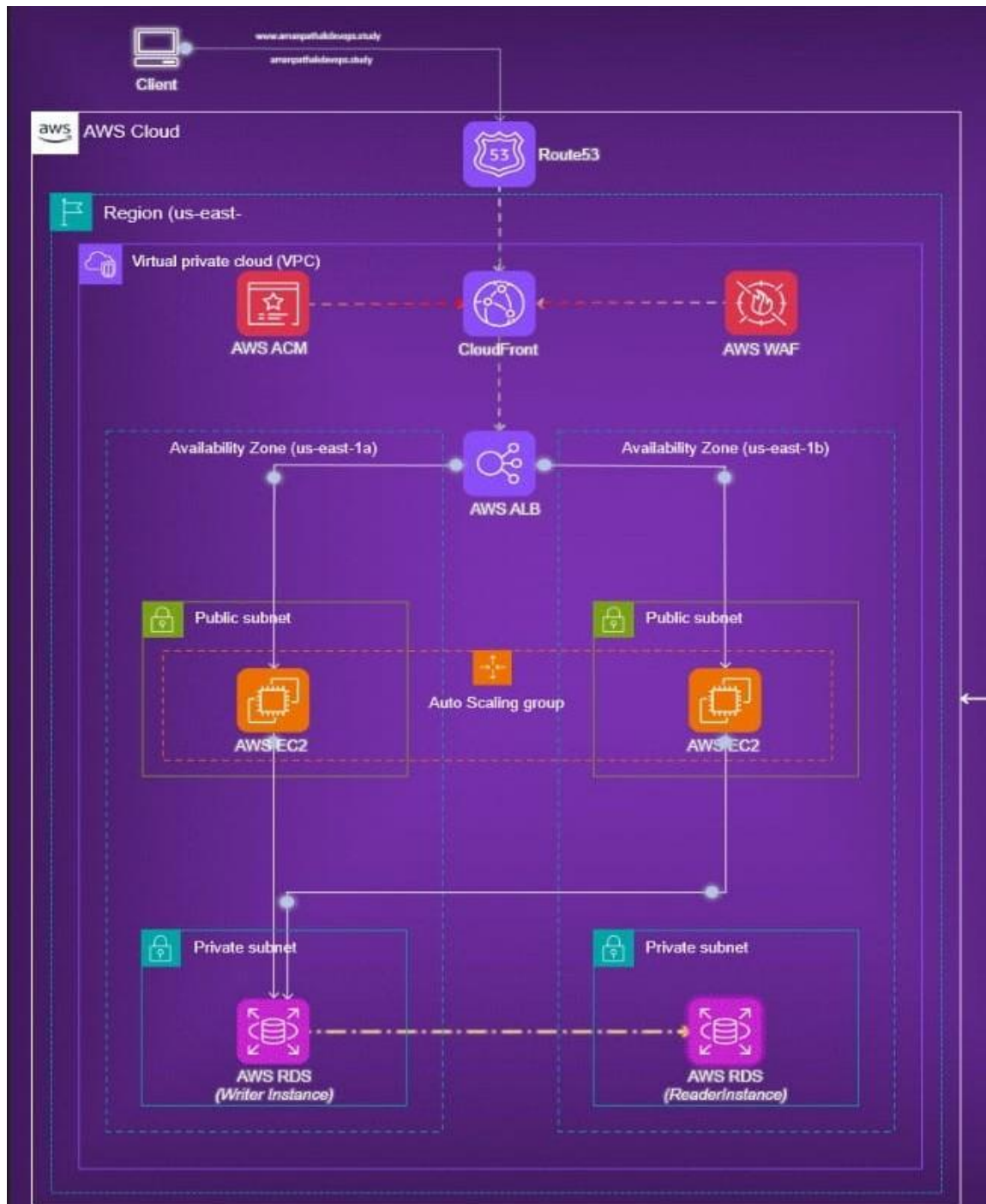


# STATIC WEB HOSTING

Name: Chakramahanti

Jawahar

Batch: 138

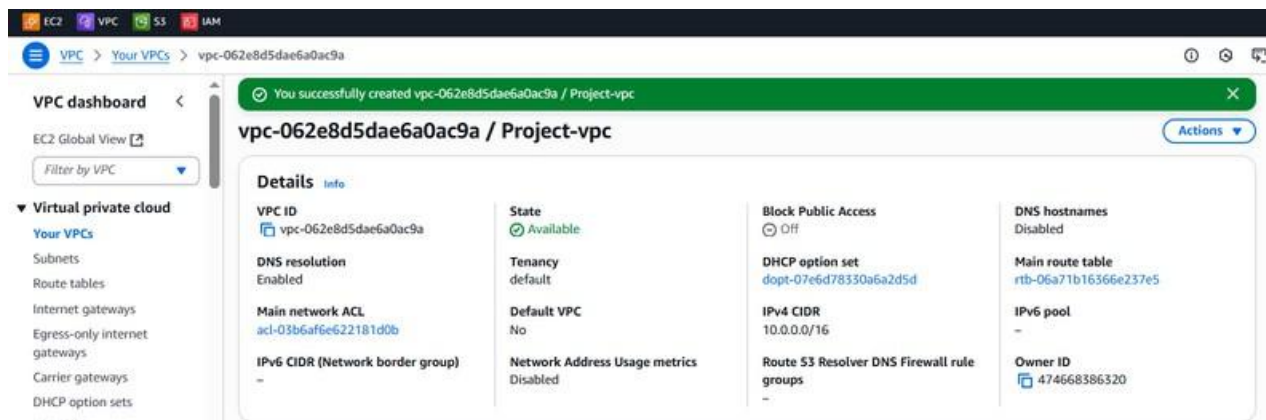


### The above setup represents a Static Web Hosting architecture on AWS:

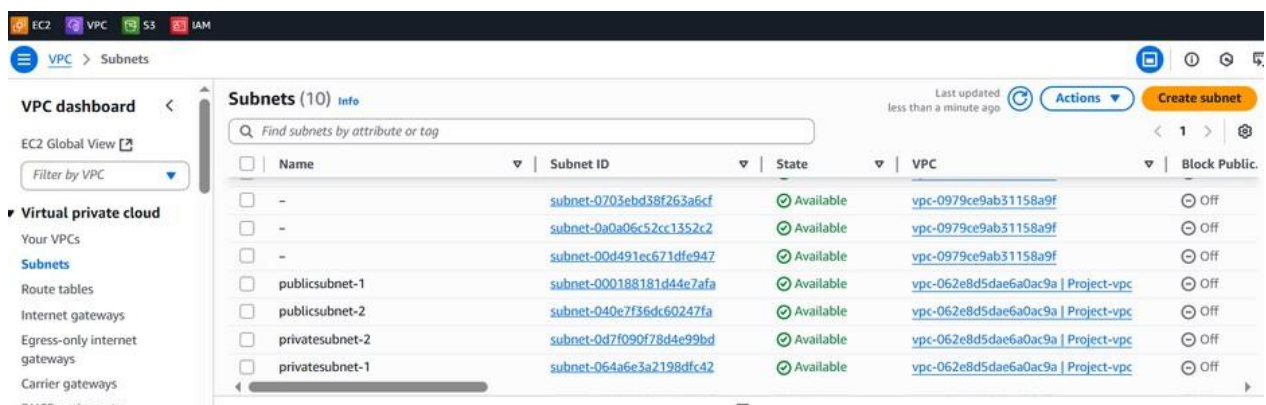
- It routes user requests via Amazon Route 53 to a scalable, secure web application that is housed on EC2 instances behind an Application Load Balancer (ALB).
- The application connects to an RDS database located in private subnets and is distributed across many Availability Zones with Auto Scaling for high availability.
- Performance, security, and HTTPS encryption are enhanced with the integration of services such as CloudFront, ACM, and WAF.

To achieve this project, the following steps were performed:

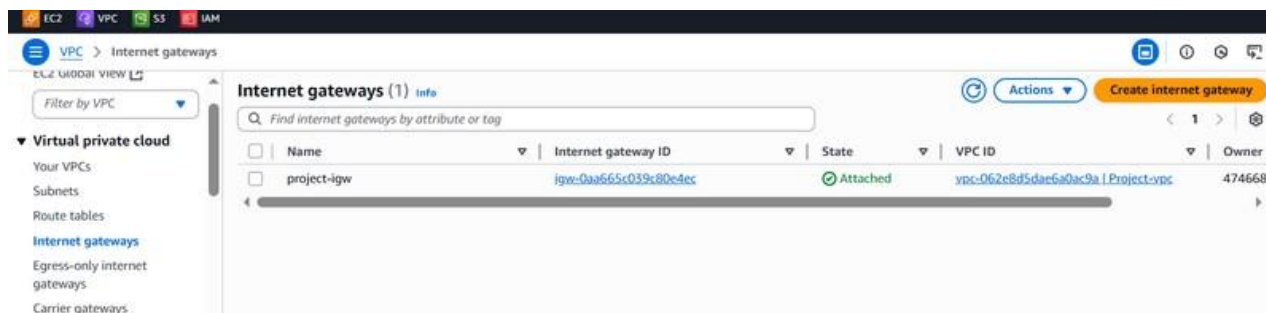
### Step 1 : Creating a VPC and its components



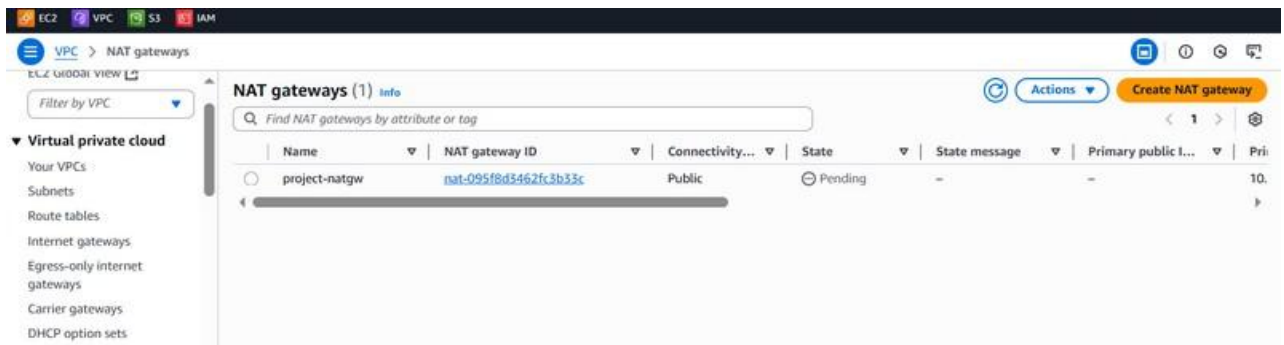
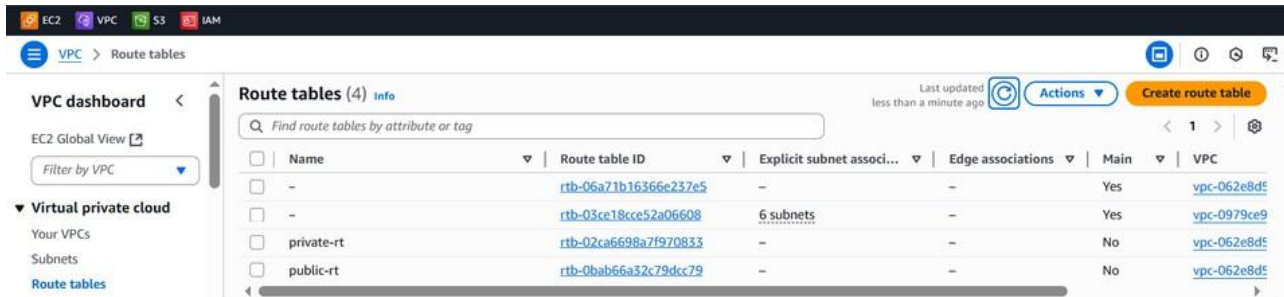
- Subnet Setup: 2 Public, 2 Private



- Internet Gateway Setup and attach to VPC.

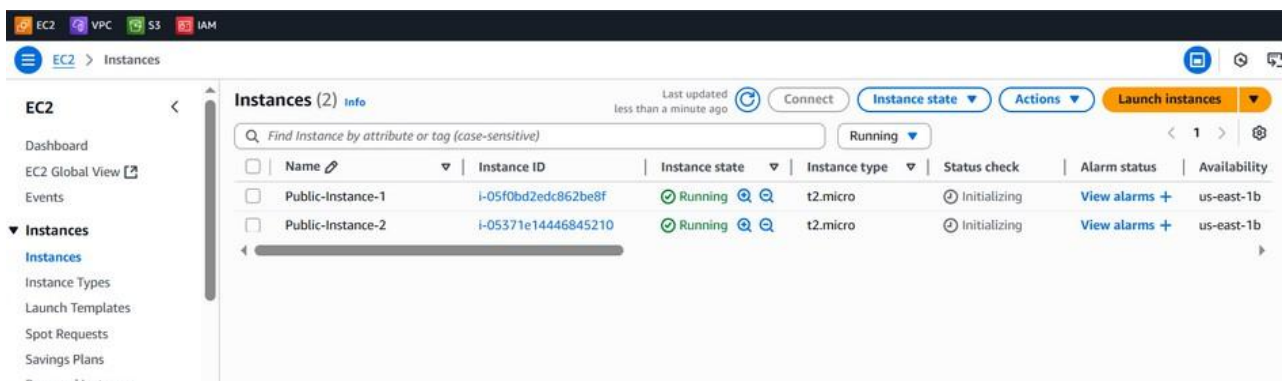


- Create Route tables and NAT Gateway



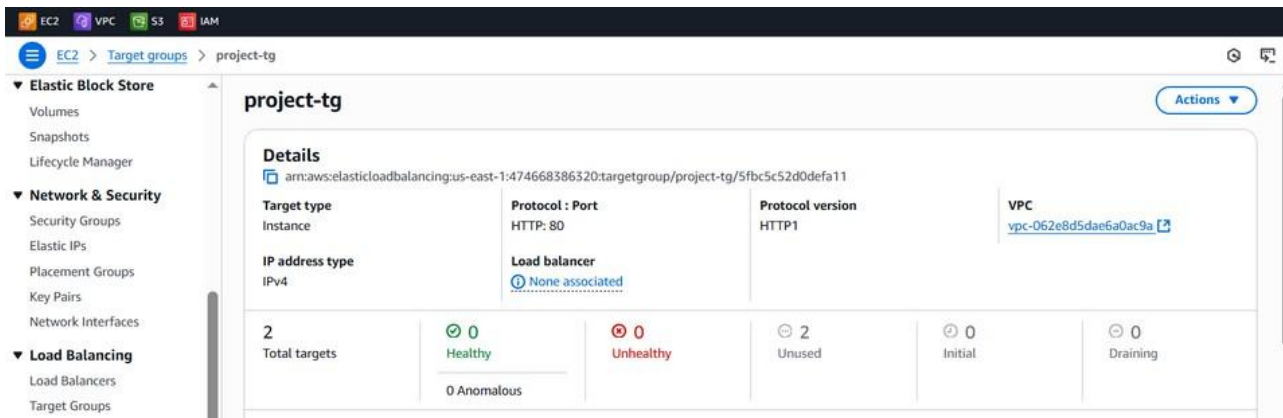
After creating the subnets, route tables are configured and associated with their respective subnets. An Internet Gateway (IGW) is attached to the public route table to enable internet access, while a NAT Gateway (NAT-GW) is connected to the private route table to allow outbound internet access for resources in private subnets.

## Step 2: Launch EC2 Instances



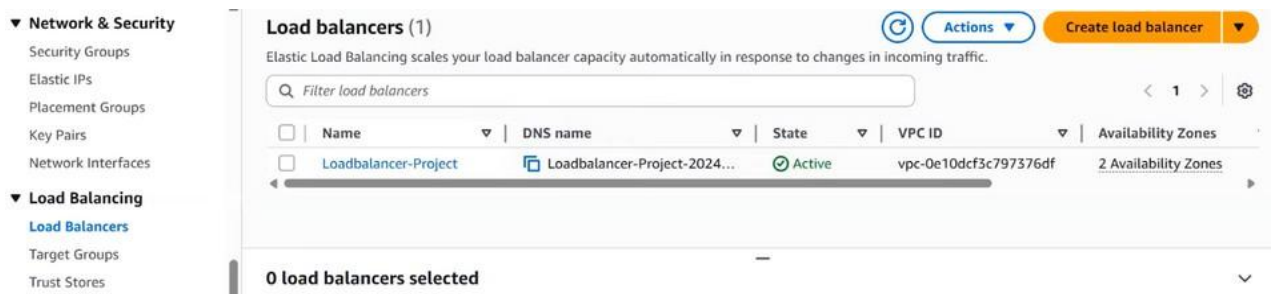
## Step 3: Create Target Group

- Create a target group named project-tg and select the appropriate VPC.
- Choose the running public instances (Public-Instance-1 and Public-Instance-2), click “Include as pending”, and then proceed to create the target group.



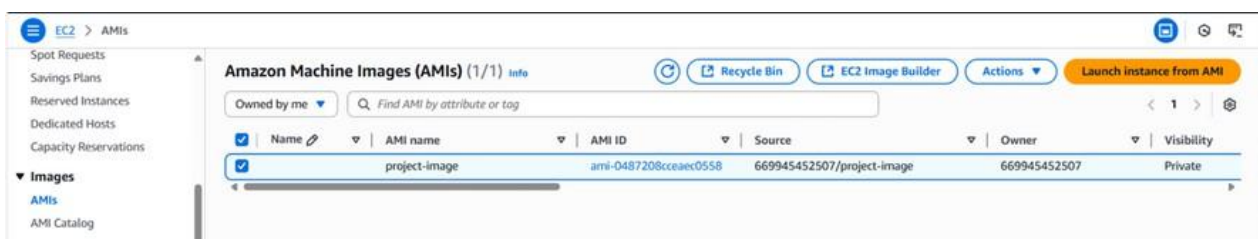
## Step 4: Create a Load Balancer

- Create an Application Load Balancer.
- Attach the previously created target group to the load balancer during configuration.
- Once created, copy the Load Balancer DNS name and paste it into the browser to access the application.
- When a request is made, the Load Balancer distributes traffic evenly between the two servers.
- Each time the page is refreshed, the request is forwarded to either server-1 or server-2 in a round-robin manner.



## Step 5: Create an AMI (image)

- After running instances , click on actions.
- Click on image template and click on image.
- Name image as project-img.
- Click create image.



## Step 6: Launch Template

EC2 > Launch templates > Create launch template

**Launch template name and description**

Launch template name - required  
Project-Template  
Must be unique to this account. Max 128 chars. No spaces or special characters like '&', "'", '@'.

Template version description  
A prod webserver for MyApp  
Max 255 chars

**Auto Scaling guidance** [Info](#)  
Select this if you intend to use this template with EC2 Auto Scaling  
☒ Provide guidance to help me set up a template that I can use with EC2 Auto Scaling

► Template tags  
► Source template

**Launch template contents**  
Specify the details of your launch template below. Leaving a field blank will result in the field not being included in the launch template.

**Summary**

Software Image (AMI)  
-

Virtual server type (instance type)  
-

Firewall (security group)  
-

Storage (volumes)  
-

Free tier: In your first year of opening an AWS account, you get

Cancel Create launch template

## Step 7: Creating Autoscaling group

- Open the Auto Scaling Group section and click “Create Auto Scaling Group”.
- Enter the name as Project-AutoScaling, select the existing launch template (Project-Template), and proceed.
- Choose the VPC, attach the existing Load Balancer, set the desired capacity to 2 and maximum to 5, then click Next.
- Finally, review the configuration and click “Create Auto Scaling Group”.

EC2 > Auto Scaling groups

Auto Scaling groups (1) [Info](#) Last updated less than a minute ago [Launch configurations](#) [Launch templates](#) [Actions](#) [Create Auto Scaling group](#)

Search your Auto Scaling groups

<input type="checkbox"/>	Name	Launch template/configuration	Instances	Status	Desired capacity	Min	Max
<input type="checkbox"/>	<a href="#">Project-AutoScaling</a>	<a href="#">Project-Template</a>   Version Default	0	Updating capacity...	2	2	3

## Step 8: Create Subnet group

- Give availability zones and select all private subnets from each zone.
- Create DB subnet group.

Aurora and RDS <

Dashboard  
Databases  
Query editor  
Performance insights  
Snapshots  
Exports in Amazon S3  
Automated backups  
Reserved instances  
Proxies

Successfully created Subnet-Group-Project. [View subnet group](#)

**Subnet groups (1)** [Refresh](#) [Edit](#) [Delete](#) [Create DB subnet group](#)

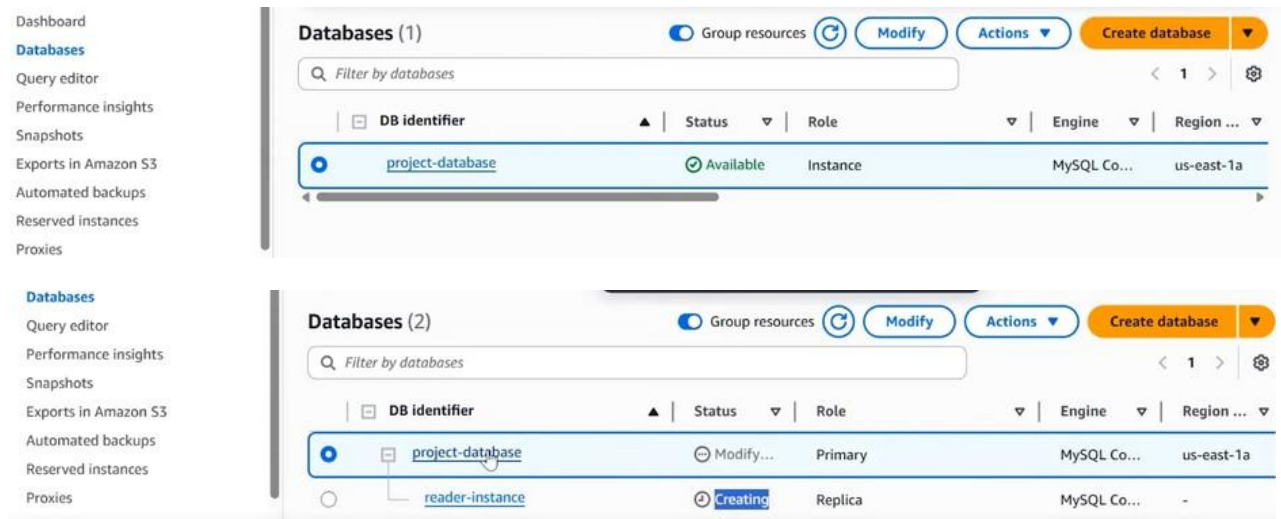
Filter by subnet group

<input type="checkbox"/>	Name	Description	Status	VPC
<input type="checkbox"/>	<a href="#">subnet-group-project</a>	allow	Complete	vpc-0e10dcf3c797376df



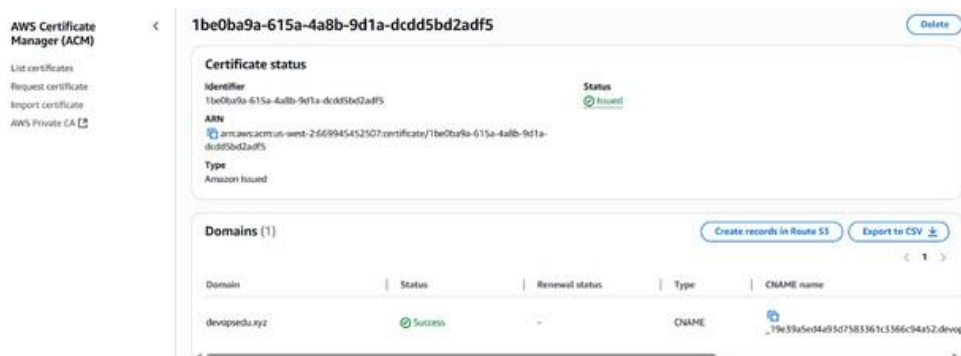
## Step 9: Create Database (RDS)

- An RDS instance was launched as the primary database (Writer), and a Read Replica was created to handle read operations.
- This setup improves performance and ensures high availability for the database layer.



## Step 10 : ACM - Request SSL Certificate

- Requested an SSL/TLS certificate in AWS Certificate Manager (ACM)



## Step 11 : Route 53 Creating hosted zones, Records

- Go to Route 53 and click on “Create Hosted Zone”.
- Enter your domain name, select the type as Public, and click “Create Hosted Zone”.
- Next, create two records:

### A Record:

Click “Create Record”, enter the subdomain as www, select the type as A, and enable Alias. Choose the endpoint as Alias to Application Load Balancer, and select your Load Balancer.

### CNAME Record:

Create another record, enable Alias, and choose the endpoint as Alias to another record in this hosted zone.

Select the previously created A record, and click “Create Record”.

**Route 53**

Public **devopsedu.xyz** Info

Hosted zone details

Records (4) | DNSSEC signing | Hosted zone tags (0)

Records (4) Info

Automatic mode is the current search behavior optimized for best filter results. [To change modes go to settings.](#)

Filter records by property or value

Record ...	Type	Routin...	Differ...	Alias	Value/Route traffic to	TTL (s...)	Health ...	Evalu...
devopsed...	A	Simple	-	Yes	dxwrcpdcqu4h5.cloudfront.net.	-	-	No
devopsed...	NS	Simple	-	No	ns-1127.awsdns-12.org. ns-980.awsdns-58.net. ns-9.awsdns-01.com. ns-1561.awsdns-03.co.uk.	172800	-	-
devopsed...	SOA	Simple	-	No	ns-1127.awsdns-12.org. aws...	900	-	-
_19e39a5...	CNAME	Simple	-	No	_8a4ee9992f28408004c544...	300	-	-

## Step 12: Creating CloudFront

Set up aCloud Front Distribution using the certificate.

Next select load balancer

Enable WAF and select your SSL certificate, then click on create distributions.

**CloudFront**

Distributions

Distributions (1) Info

Search all distributions

All distributions

ID	Status	Description	Type
E17OMQ43PQZM38	Enabled	-	Standard

Next add this distribution in Route 53 records. secure communication.

Added an HTTPS listener to the Load Balancer.

**Add listener** Info

Add a listener to your Application Load Balancer (ALB) to define how client requests and network traffic are routed within your application. Every listener is made up of a default action that's required and can only be edited. Additional rules can be added, edited and deleted from the listener.

Load balancer details: project-alb

**Listener: HTTPS:443**

A listener checks for connection requests using the protocol and port that you configure. The default action and any additional rules that you create determine how the Application Load Balancer routes requests to its registered targets.

**Listener configuration**

The listener will be identified by the protocol and port.

**Protocol**

Used for connections from clients to the load balancer.

HTTPS

**Port**

The port on which the load balancer is listening for connections.

443

1-65535

**Default action** Info

The default action is used if no other rules apply. Choose the default action for traffic on this listener.

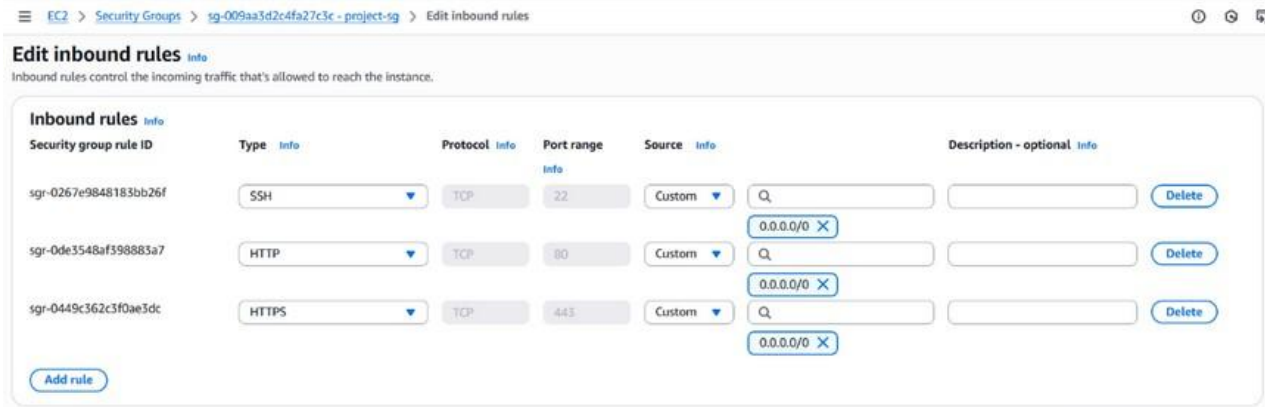
**Authentication action - optional** Info

Authentication requires IPv4 connectivity to authentication endpoints. [Learn more](#)

☐ **Authenticate users**

Configure user authentication through either OpenID Connect (OIDC) or Amazon Cognito.

We have to add HTTPS route to the Load Balancer Security Group.



### Step 13: Testing and Verification

- Copy your domain and paste it in browser you will see lock symbol before the domain and it is HTTPS, because of ACM our connection is secure now.

